

A Real-Time Flight Safety Alarm System for Multiple UAVs

Khin Thida San^{1,a} and Yoon Seok Chang¹

¹UTAC, School of Air Transportation and Logistics, Korea Aerospace University, Goyang-si, Seoul 412-791, South Korea

Abstract. Airborne heterogeneous UAVs can encounter flight control or sensor failures while performing flight tasks. To protect against this, safety alarm systems (SAS) are typically implemented in ground control systems (GCS) to prevent crashes or other failures. Tracing the failure for a single UAV is relatively easy, but it is difficult to manually monitor multiple UAVs flying in a distributed open area. The current paper aims to fulfil this SAS deficiency, a deficiency that is still frequently ignored during GCS development. Our SAS alerts users of no-fly zones as well as navigation, battery, GPS and communication failures. The system can monitor simultaneous multiple UAV-flight failures in real time so that GCS can send appropriate commands to each UAV encountering a problem.

1 Introduction

UAVs are increasingly being applied in diverse fields. Logistics companies use them to carry out deliveries; large-scale corporations and environmental agencies use them to monitor assets such as facilities, gas pipelines, crops and forests; and the police use them for surveillance and traffic control. Typical UAV flight missions should prioritize consistency and safety.

In particular, they should endeavor to (1) maintain navigation consistency in terms of matching flight inertial measurement units (IMU) with the global positioning system (GPS) to arrive at the desired location, (2) achieve communication consistency between the UAV and GCS for monitoring purposes, (3) conserve energy consistency to guarantee flight times for desired tasks and, finally, (4) avoid no-fly zones and obstacles as shown in Fig. 1.

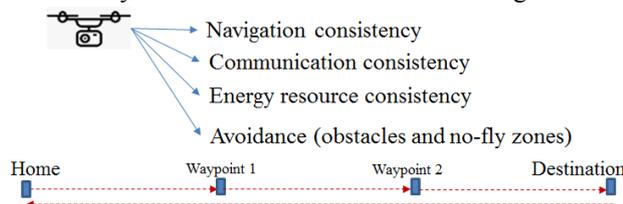


Figure 1. UAV safety consistency factors.

Users must establish an effective GCS to control and monitor UAV flights. There are several open-source GCS, including Mission Planner and QGroundControl, which can be easily downloaded from the internet. Most drone vendors provide their own GCS, DJI being a prime example. However, there is still a lack of awareness regarding SAS. As such, this paper begins by delving into the background of flight security issues and SAS implementation to elicit better flight safety.

^a Corresponding author: khinthidasan.pont@gmail.com

2 Background

To overcome traditional communication line-of-sight (LOS) limitations, a modern GCS can now stay in communication with UAVs over long distances via LTE communication. Most urban regions and much of the countryside are already covered by mobile networks. In line with the Internet-of-Things (IoT) concept, if a UAV can link itself to the internet via a mobile network, area limitations become virtually non-existent [1-2]. For example, in Qatar, an autopilot system was run via Raspberry Pi with a 4G LTE modem to transmit air pollution sensor data to a base station in semi-real time [1]. (This system almost achieved real-time but encountered a three- to four-second delay for the video transmission. Data transmissions without video are naturally faster.) Another autopilot option is to use Pixhawk, which either connects to Raspberry Pi with a serial connection [3]. The DroneKit API, which runs on a Linux environment with a Raspberry Pi computer can obtain telemetry data from Pixhawk and convey them directly to the internet. In this case, the UAV itself can communicate to the GCS from anywhere within the boundaries of the 4G LTE network. Collaborative UAVs can be monitored via the cloud using the Web-of-Things (WoT) concept, which supports the monitoring of multiple UAVs anywhere simultaneously [4-6]. Each UAV connects to the cloud, where the cloud server keeps all telemetry data, and each UAV can retrieve commands from the GCS as well.

For safety, most commercial drones use common fail-safe systems such as Return to Launch (RTL) or Land at the Current Spot (LAND) when the UAV invades the geo-fence or battery failsafe. A UAV can save its home

position in its flight memory, and it will RTL when it detects the boundary of the predefined geo-fencing area. Likewise, when a UAV detects a low battery level, it can LAND or RTL. However, these failsafes are insufficient on long missions. Moreover, the geo-fencing failsafe becomes impractical when there are several no-fly zones impeding on the flight missions.

Drone strikes have become a major issue all over the world. If a UAV is sucked into an airplane engine, its Li-po battery can produce a powerful explosion. Furthermore, UAVs could potentially be used by criminals or terrorists and create safety and security problems. For these reasons, many countries strictly regulate UAV flights in zones near airports, in heavily populated areas and near military installations. UAVs typically navigate positions by following a GPS signal received by its GPS device. The no-fly zones normally transmit frequency noises to disturb the GPS signal from the satellite or control signal from the UAV operator. Drone jammers, which are essentially signal blockers, are used to block and interfere with a UAV's GPS signals [7]. If a UAV cannot listen to signals due to frequency noises and does not know where to go, it will crash or fly at the same spot until it depletes its battery. Therefore, it is essential to avoid no-fly zones.

The geo-fencing algorithms for polygonal and circular areas are implemented in the GCS [8]. The algorithms sense whether the UAV is inside or outside of the geo-fence and has an alarm to warn the operator. This geo-fencing concept can be assumed as containing no-fly zones. For research efficiency, our system not only warns against flying into no-fly zones, but also detects approaching zones in real time with respect to UAV heading and distance. Moreover, miss-headings and miss-routings are detected to trace navigation failures due to strong winds. After the SAS has detected the drifting error during navigation, the error can then be corrected by determining the correct crab angle and saving the heading to a predefined path [9-10].

Powerful UAV control systems become crucial when multiple UAVs are operated simultaneously. GCS implementation for multiple UAVs has been carried out in numerous studies [11-13]. Another study proposed the implementation of a monitoring system for UAVs delivering products [14]. These studies relied on open source libraries and APIs to receive flight data and control the UAVs. An SAS should be imbedded as an additional safety module for all types of GCS for UAV safety.

3 System interface and implementation

The system interface is the translation layer between the physical UAV and SAS, as shown in Figure 8. A MAVLink header protocol is used to receive flight data from the drone. Meanwhile, Drone-Kit Python [15] is used to translate the protocol into readable data form. The real-time flight data to feed the SAS are shown in Fig. 2. System ID is used to identify specific UAVs separately. Satellite visibility numbers are used to check the GPS signal strength. (There are currently 24 GPS satellites in

Earth's orbit, and at least 4 satellites are needed for accurate positioning [16]). The battery capacity is monitored for the battery alarm (The Li-po battery can explode or be of no use if the battery drops to 0%. Therefore, 20% is set as the bottom margin before a recharge is required). Heading or bearing degree and current position are used together to trace navigation failures and approaching no-fly zones. The next waypoint is the current heading waypoint among a series of predefined waypoints. It is used to trace navigation failures. MySQL database is used to save all flight data. C#, JavaScript and the Google Maps API are used for GCS development. SAS is established in the GCS as a module.

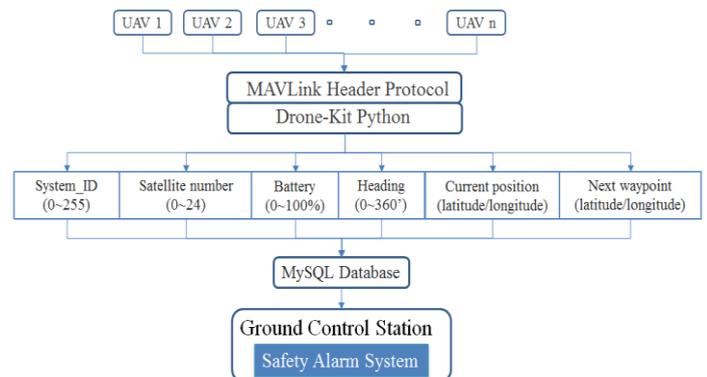


Figure 2. System interface via Drone-Kit python and implementation block diagram.

4 Safety alarm system (SAS)

Table 1. The alarms and their purposes.

No	Alarm	Purpose
1	Miss-heading alarm	To detect flight navigation failure
2	GPS alarm	To detect flight navigation failure
3	Area approaching alarm	To perform no-fly zones avoidance
4	Area invading alarm	To perform no-fly zones avoidance
5	Connection alarm	To detect communication failure
6	Battery alarm	To detect battery capacity

The SAS is designed to alert operators with its own threshold levels and threshold timing/unit, which vary according to system user input. In every period of threshold timing, the threshold level of each UAV is checked. All alarms are attentive and inattentive automatically according to their threshold levels as compared to the real-time levels from the UAV.

4.1 Miss-heading alarm

The miss-heading alarm detects flight navigation failures. The UAV navigates via predefined waypoints. The next heading waypoint and real-time heading data are used to detect this alarm. The initial bearing is 129 degrees from Waypoint 1 (Latitude: 37.59789055823786, Longitude:

126.86368900094021) to Waypoint 2 (Latitude: 37.59744852641662, Longitude: 126.86437564644802), as shown in Fig. 3(left). Normally, the real-time heading for a normal flight without navigation errors will keep between 128 and 129 degrees as its heading values. Therefore, a \pm threshold degree of initial bearing can be used to alert for miss-headings.

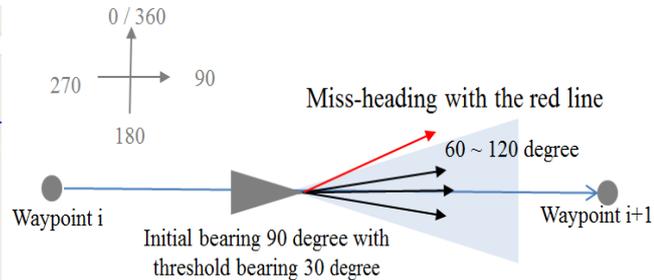
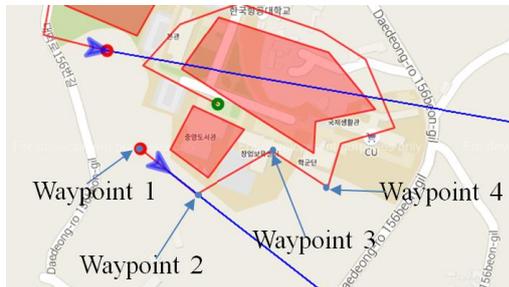


Figure 3. UAV flight with initial heading of 129 degrees between Waypoints 1 and 2 (left), and miss-heading detection via the heading/bearing angle degree (right).

4.2 GPS alarm

The GPS alarm helps prevent flight navigation failures. The autonomous UAV follows the GPS signal to reach its desired position. The accuracy of the satellite positioning depends on the satellite visibility number. The alarm will ring if the current satellite number is lower than the threshold level.

4.3 Area approaching alarm

The area approaching alarm enables the operator to avoid restricted areas or no-fly zones. The real time heading data together with the current position are used. For example, if the threshold distance is 0.5 km, every 1-meter point along the 0.5 km is checked to determine whether the no-fly zone's coordinates were touched. Point-In-Polygon (PIP) problems via a ray casting algorithm [17] is applied to detect the heading line intersection. If there is any number (even or odd) of intersections for a passing ray, the heading is approaching.



Figure 4. UAV heading with the blue line (0.5-km detection) used for the approaching alarm.

4.4 Area invading alarm

The area invading alarm alerts the GCS when the UAV has passed into restricted areas or no-fly zones. The SAS saves a series of coordinates for the no-fly zone polygons. In real time, the current position is checked to determine

For example as shown in Fig. 3(right), when Waypoint (i) to (i+1) has an initial bearing of 90 degrees, the threshold level is ± 30 degrees and the threshold time is 5 seconds. The alarm will ring if the current bearing degree is over the \pm threshold degree (60 to 120 degrees) for 5 seconds. The alarm will deactivate once the drifting degree normalizes.

whether the UAV is inside or outside of the polygons by using the ray casting algorithm [17]. An odd number of passing ray intersections indicates that the UAV is inside, while an even number indicates outside.



Figure 5. Area invading UAV.

4.5 Connection alarm

The connection alarm detects communication failures between the UAV and GCS (SAS). The UAV can still fly even if it loses its GCS connection because the predefined GPS waypoints can still guide the UAV. However, GCS communication is still important as it facilitates additional commands (e.g. LAND, RTL) to the UAV when encountering dynamic environment changes such as sudden rain or wind. The real-time data is received every second from the UAV.

4.6 Battery alarm

The battery alarm detects battery failures. It is simple and easy to implement. If the real-time battery level of the flight message drops under the predefined threshold level, the alarm will ring.

5 Flight failure compensation

The SAS compensation nature is semi-autonomous where the operator decision is still required at some level of the failures.

1. Miss heading - When this alarm is triggered, SAS sends RTL to UAV to fly back to the home position at once to avoid unnecessary flight failure.

2. GPS - Even though this alarm is triggered, SAS lets UAV to keep flying until miss-heading alarm is shown up. Or, the operator can cancel the mission via RTL alternatively.

3. Area approaching - Even though this alarm is triggered, UAV is still outside of the no-fly zones. The operator must be aware of UAV current situation. SAS lets UAV to keep flying until miss-heading alarm is shown up. Or, the operator can cancel the mission via RTL alternatively.

4. Area invading - When this alarm is triggered, SAS sends RTL to UAV to fly back to the home position at once to avoid unnecessary flight failure.

5. Connection - Typically, UAV can fly by itself via its GPS navigation even it loses the communication to GCS. In this case, SAS cannot control UAV because of the communication failure. Therefore, UAV should be set RTL as a failsafe before flight. On the way back, if the communication reconnects again, the operator can decide UAV to fly via its GPS navigation (without concerning communication loss) or completely cancel the mission.

6. Battery - The distance between UAV and home place is considered. If the current battery is enough to fly back to home place, RTL mode is activated by SAS automatically, if the battery level is insufficient, SAS will alert the operator to find the landing point from the real-time video feeding manually.

6 Experimental results

The experiment was carried out via a Software-In-The-Loop (SITL) simulation [18] provided by Drone-Kit Python. SITL supports the Pixhawk autopilot with a MAVLink messaging system for the developer to attain a real-flight environment. However, SITL supports only one UAV at a time. To facilitate multiple UAVs flying simultaneously in the test bed, the SITL ran three flight missions one by one and saved them separately. Those 3 missions were run sequentially to simulate multiple UAV flights when the real experiment started. The experiment is shown in Fig. 6. UAVs 1, 2, 3, 4 and 5 were used, and their status messages were received in status message form. We assumed that UAVs 4 and 5 were on the ground in landing state, though they were alive and sending flight data. UAVs 1 and 2 were in flying state under AUTO mode (waypoint navigation mode). UAV 3 was in GUIDED mode (i.e. the mode receiving external commands) to force it to go inside of the no-fly zone. All alarm messages for all related UAVs with their thresholds and real time units can be seen in alarm form in Fig. 6.

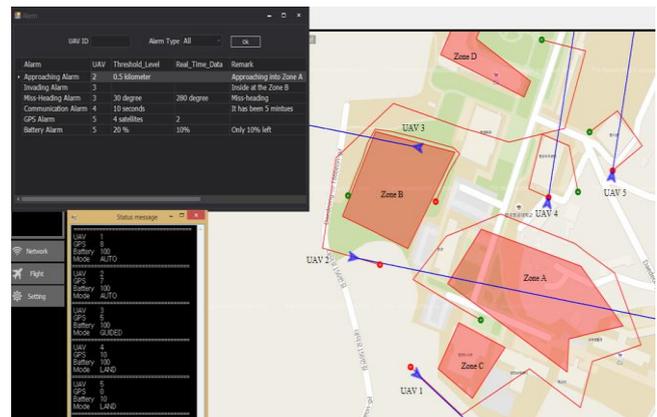


Figure 6. SAS facilitating multiple flights

7 Conclusions

Our SAS proposed an improved safety concept to alert operators of UAV flight failures stemming from internal sensor failures or external environmental effects. According to the situation and issue level, the GCS compensated for changes by sending urgent commands to either change UAV flight routes, fly back to base or conduct an emergency landing. In the future research, we will implement the autonomous compensation method for the navigation failures of miss-heading awareness and no-fly zones awareness without the contribution of the operator decision. We hope to implement an obstacle avoidance alarm system via object detection over real-time video streaming to alert the GCS that the UAV is approaching obstacles.

Acknowledgements

This research was supported by Ministry of Science and ICT (MSIT), Korea, under Information Technology Research Center (ITRC) support program (IITP-2017-2014-0-00678) supervised by Institute for Information & communications Technology Promotion (IITP).

References

1. K. Al-Hajjaji, M. Ezzin, H. Khamdan, and A.E. Hassani. Design, development and evaluation of a UAV to study air quality in Qatar. *College of Engineering, Qatar University* (2017)
2. J.W. Park, H.H. Jeong, J.S. Kim, and C.U. Choi. Development of open source-based automatic shooting and processing UAV imagery for orthoimage using smart camera UAV. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B7: 941-944 (2016)
3. <http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html>
4. S. Mahmoud, and N. Mohamed. Collaborative uavs cloud. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. pp. 365-373 (2014)

5. S. Mahmoud, N. Mohamed, and J. Al-Jaroodi. Integrating UAVs into the cloud using the concept of the Web of Things. *Journal of Robotics*. Vol 2015 (2015)
6. C. Honh, and D. Shi. A control system architecture with cloud platform for multi-UAV surveillance. *2018 IEEE (SmartWorld /SCALCOM /UIC /ATC /CBDCOM /IOP/SCI)* (2018)
7. A. Perkins, L. Dressel, S. Lo, T. Reid, K. Gunning, and P. Enge. Demonstration of UAV-based GPS jammer localization during a live interference exercise. *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS)*. pp. 3094-3106 (2016)
8. P.L. Pratyusha, and V.P.S. Naidu. Geo-fencing for unmanned aerial vehicle. *International Journal of Computer Applications National Conference Electronics, Signals, Communication and Optimization (NCESCO 2015)*. pp. 0975-8887 (2015)
9. A.S. Biradar. Wind estimation and effects of wind on waypoint navigation of UAVs (2014)
10. F. Schiano, J. Alonso-Mora, K. Rudin, P. Beardsley, R. Siegwart, and B. Siciliano. Towards estimation and correction of wind effects on a quadrotor UAV. *International Micro Aerial Vehicle Conference (IMAV)*. pp. 12-15 (2014)
11. D. Perez, I. Maza, F. Caballero, D. Scarlatti, E. Casado, and A. Ollero. A ground control station for a multi-UAV surveillance system. *Journal of Intelligent and Robotic Systems*. **69**, pp. 119-130 (2013)
12. M. Dong, B.M. Chen, G. Cai, and K. Peng. Development of a Real-time onboard and ground station software system for a UAV helicopter. *Journal of Aerospace Computing, Information and Communication*. **4**, pp. 933-955 (2007)
13. G. Zhang, X. Lei, Y. Niu, and D. Zhang. Architecture design and performance analysis of supervisory control system of multiple UAVs. *Defense Science Journal*. **65(2)**, pp. 93-98 (2015)
14. K.T. San, S.J. Mun, Y.H. Choe, and Y.S. Chang. UAV delivery monitoring system. *Asia Conference on Mechanical and Aerospace Engineering (ACMAE 2017)*. **151** (2017)
15. <http://python.dronekit.io/>
16. <https://medium.com/iotforall/6-things-you-didnt-know-about-gps-efb37e83bd02>
17. https://en.wikipedia.org/wiki/Point_in_polygon#Ray_casting_algorithm
18. http://python.dronekit.io/develop/sitl_setup.html